# CIT 591 Introduction to Software Development

## Module 9: Exception Handling

## Module Learning Objectives

- Create more robust programs with exception handling
- Anticipate and design to handle unexpected user inputs

## Module Glossary

- **Try-catch block:** A statement for handling exceptions. Place the statement into a location of your program that knows how to handle a particular exception. The try block contains one or more statements that may cause an exception of the kind that you are willing to handle. Each catch clause contains the handler for an exception type.

## Module Resources

- **Textbook Readings:**
  - 11.4: Exception Handling
  - 11.5: Application: Handling Input Errors
- **Websites:**
  - finally blocks (Java Documentation)
  - Throwing exceptions is like a relay throw in baseball
  - Throwing exceptions is like a relay throw in cricket

## Key Concepts & Examples

**Handling Exceptional Situations:**

There are two ways of dealing with exceptions

a) put your code in a try block. Follow that block up with a catch block where you handle the exception.

**b)** accept that your code will throw an exception and pass the buck - done by adding a `throws` to your method

- Examples used in the video: What happens when a user is supposed to giving you integer input but accidentally enters a character

**Java's Exception Hierarchy:**
Java has checked exceptions and unchecked exceptions. A checked exception is something that Java insists that you deal with, e.g. FileNotFoundException. There are lots of in-built exception types in Java, so don't reinvent the wheel! Get used to using/extending existing exceptions. Be sure to reference the documentation here for a list of in-built exception types.

- Examples used in the video: Exception handling being added to the PositiveInteger example from assignment 2.